

# Manajemen Sistem File

# Interface Sistem File

- **Konsep Berkas**
- Berkas adalah sebuah koleksi informasi berkaitan yang diberi nama dan disimpan di dalam *secondary storage*. Biasanya sebuah berkas merepresentasikan data atau program. Ada pun jenis-jenis dari berkas:
- *Text file*: yaitu urutan dari karakter-karakter yang diatur menjadi barisan dan mungkin halaman.
- *Source file*: yaitu urutan dari berbagai subroutine dan fungsi yang masing-masing kemudian diatur sebagai deklarasi-deklarasi diikuti oleh pernyataan-pernyataan yang dapat diexecute.
- *Object file*: yaitu urutan dari byte-byte yang diatur menjadi blok-blok yang dapat dipahami oleh penghubung system.
- *Executable file*: adalah kumpulan dari bagian-bagian kode yang dapat dibawa ke memori dan dijalankan oleh loader.

# Interface Sistem File

- **Metode Akses**

- Berkas menyimpan informasi. Apabila sedang digunakan informasi ini harus diakses dan dibaca melalui memori komputer. Informasi dalam berkas dapat diakses dengan beberapa cara. Berikut adalah beberapa caranya:
- **Akses Sekuensial** : Akses ini merupakan yang paling sederhana dan paling umum digunakan. Informasi di dalam berkas diproses secara berurutan. Sebagai contoh, editor dan kompilator biasanya mengakses berkas dengan cara ini.
- **Akses Langsung** : Metode berikutnya adalah akses langsung atau dapat disebut *relative access*. Sebuah berkas dibuat dari rekaman-rekaman logical yang panjangnya sudah ditentukan, yang mengizinkan program untuk membaca dan menulis rekaman secara cepat tanpa urutan tertentu.

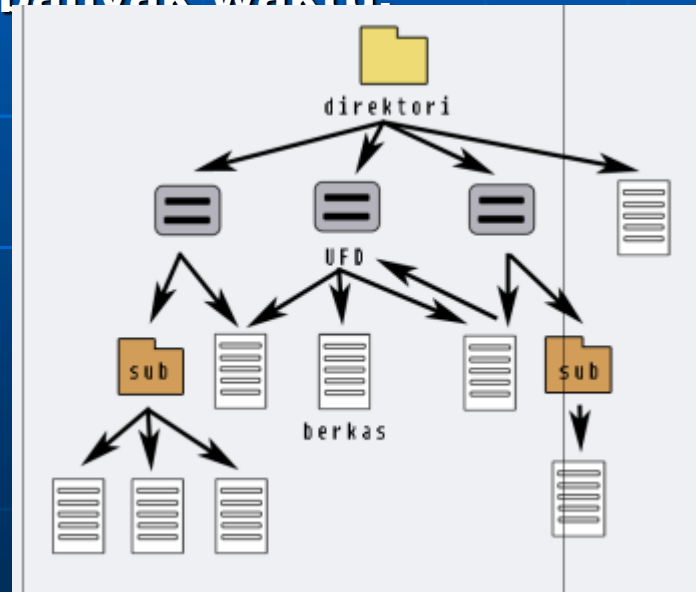
# Struktur Direktori

## Operasi Direktori

- Mencari Berkas
- Membuat berkas
- Menghapus berkas
- Menampillkan isi direktori
- Mengubah nama berkas
- Akses Sistem berkas

# Struktur Direktori

- **Direktori dengan Struktur Graf Umum**
- **Skema ini menyangkut memeriksa seluruh sistem berkas dengan menandai tiap berkas yang dapat diakses. Kemudian mengumpulkan apa pun yang tidak ditandai pada tempat yang kosong. Hal ini tentunya dapat menghabiskan banyak waktu.**



# Proteksi

- **Proteksi**

- Dalam pembahasan mengenai proteksi berkas, kita akan berbicara lebih mengenai sisi keamanan dan mekanisme bagaimana menjaga keutuhan suatu berkas dari gangguan akses luar yang tidak dikehendaki. Sebagai contoh bayangkan saja Anda berada di suatu kelompok kerja dimana masing-masing staf kerja disediakan komputer dan mereka saling terhubung membentuk suatu jaringan; sehingga setiap pekerjaan/dokumen/ berkas dapat dibagi-bagikan ke semua pengguna dalam jaringan tersebut. Misalkan lagi Anda harus menyerahkan berkas RAHASIA.txt ke atasan Anda, dalam hal ini Anda harus menjamin bahwa isi berkas tersebut tidak boleh diketahui oleh staf kerja lain apalagi sampai dimodifikasi oleh orang yang tidak berwenang. Suatu mekanisme pengamanan berkas mutlak diperlukan dengan memberikan batasan akses ke setiap pengguna terhadap berkas tertentu.



# Proteksi

- Pendekatan Pengamanan Lainnya
- Salah satu pendekatan lain terhadap masalah proteksi adalah dengan memberikan sebuah kata kunci (*password*) ke setiap berkas. Jika kata-kata kunci tersebut dipilih secara acak dan sering diganti, pendekatan ini sangatlah efektif sebab membatasi akses ke suatu berkas hanya diperuntukkan bagi pengguna yang mengetahui kata kunci tersebut.
- Meski pun demikian, pendekatan ini memiliki beberapa kekurangan, diantaranya: Kata kunci yang perlu diingat oleh pengguna akan semakin banyak, sehingga membuatnya menjadi tidak praktis.
- Jika hanya satu kata kunci yang digunakan di semua berkas, maka jika sekali kata kunci itu diketahui oleh orang lain, orang tersebut dapat dengan mudah mengakses semua berkas lainnya. Beberapa sistem (contoh: TOPS-20) memungkinkan seorang pengguna untuk memasukkan sebuah kata kunci dengan suatu subdirektori untuk menghadapi masalah ini, bukan dengan satu berkas tertentu.
- Umumnya, hanya satu kata kunci yang diasosiasikan dengan semua berkas lain. Sehingga, pengamanan hanya menjadi semua-atau-tidak sama sekali. Untuk mendukung pengamanan pada tingkat yang lebih mendetail, kita harus menggunakan banyak kata kunci.

# Implementasi Sistem File

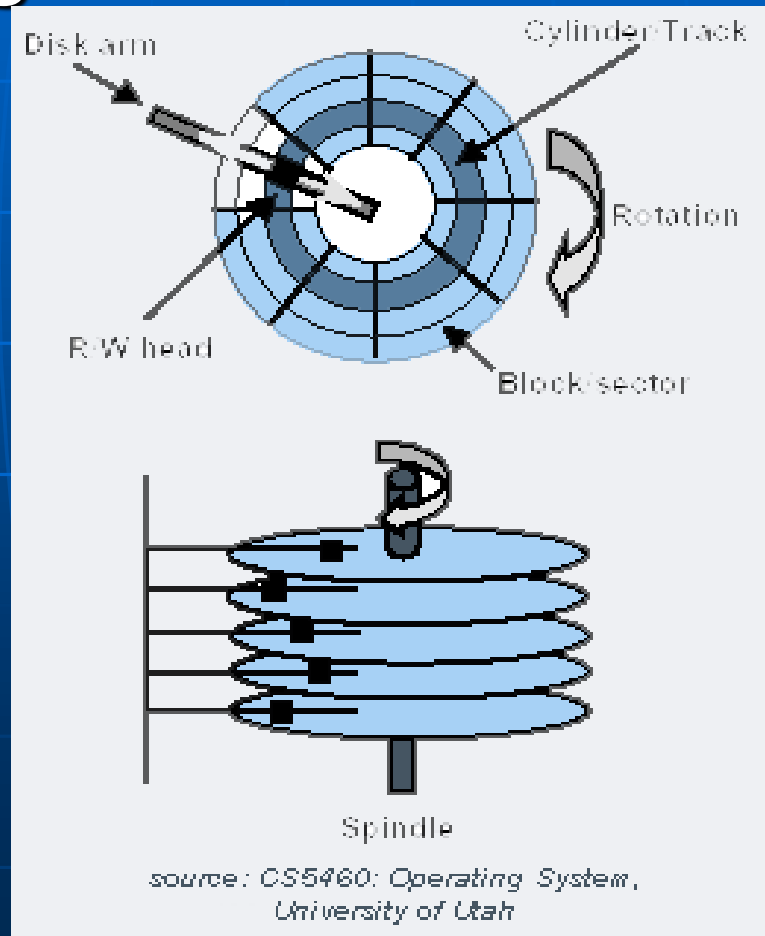
- **Struktur Sistem Berkas**

- *Disk* yang merupakan tempat terdapatnya sistem berkas menyediakan sebagian besar tempat penyimpanan dimana sistem berkas akan dikelola. *Disk* memiliki dua karakteristik penting yang menjadikan *disk* sebagai media yang tepat untuk menyimpan berbagai macam berkas, yaitu:
  - Data dapat ditulis ulang di *disk* tersebut, hal ini memungkinkan untuk membaca, memodifikasi, dan menulis di *disk* tersebut.
  - Dapat diakses langsung ke setiap blok di *disk*. Hal ini memudahkan untuk mengakses setiap berkas baik secara berurut mau pun tidak berurut, dan berpindah dari satu berkas ke berkas lain dengan hanya mengangkat *head disk* dan menunggu *disk* berputar.



# Implementasi Sistem File

- Disk Organization



# Metode Alokasi

- ***Contiguous Allocation***
- ***Linked Allocation***
- ***Indexed Allocation***

# Manajemen Ruang Kosong

- ***Bit Vector***
- ***Linked List***
- ***Grouping***
- ***Counting***

# Implementasi Direktori

- Direktori pada CP/M
- Direktori pada MS-DOS
- Direktori pada UNIX

# Efisiensi dan Kinerja

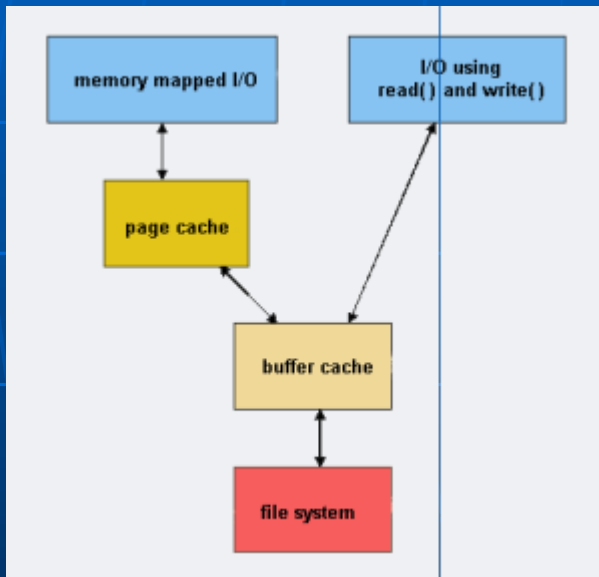
- **Efisiensi**

Penggunaan yang efisien dari ruang disk sangat tergantung pada alokasi disk dan algoritma direktori yang digunakan

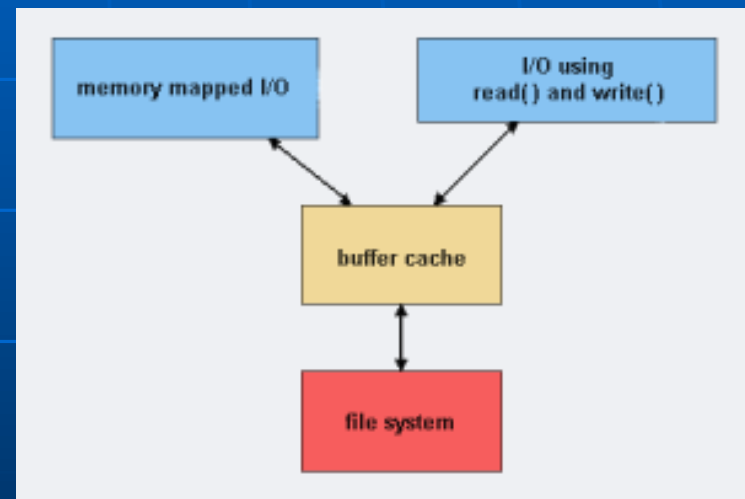
- **Kinerja**

Sekali algoritma sistem berkas dipilih, kita tetap dapat mengembangkan kinerja dengan beberapa cara. Kebanyakan dari *disk controller* mempunyai memori lokal untuk membuat *on-board cache* yang cukup besar untuk menyimpan seluruh *tracks* dengan sekejap.

# Efisiensi dan Kinerja



Tanpa *unified buffer cache*

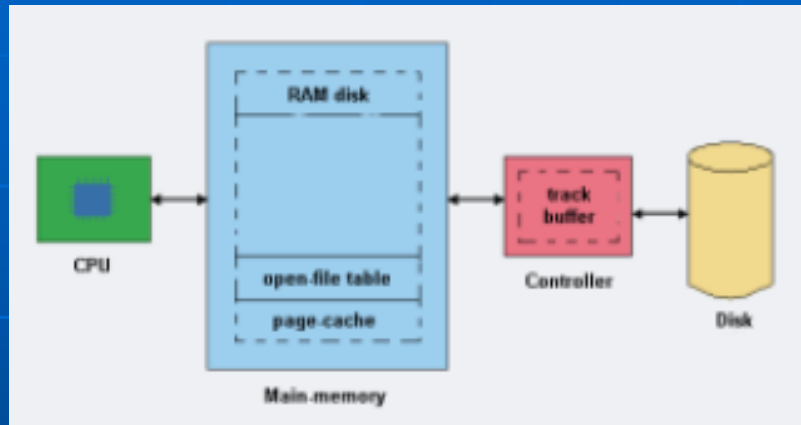


Menggunakan *unified buffer cache*



# Recovery

## ■ Pengecekan Rutin



### *Backup dan Restore*

Dikarenakan disk magnetik kadang-kadang gagal, perawatan harus dijalankan untuk memastikan data tidak hilang selamanya. Oleh karena itu, program sistem dapat digunakan untuk *back up* data dari disk menuju ke media penyimpanan yang lainnya, seperti sebuah *floppy disk*, tape magnetik, atau disk optikal. *Recovery* dari kehilangan sebuah berkas individu, atau seluruh disk, mungkin menjadi masalah dari *restoring* data dari *backup*.